|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | |  | | |
| Project: | | Helios Home Automation System | | |
| Team: | | Helios | | |
| Class: | | CSE 4392-002; Spring 2016 | | |
| Module: | | Design 2 | | |
| Deliverable: | | Design 2 | | |
| Version: | | [2.0] | Date: | [04/08/2016] |

Contributors:

Arun Kalahasti, John Green

Revision History

| Version number | Date | Originator | Reason for change | High level description of changes |
| --- | --- | --- | --- | --- |
| 1.0 | 02/26/2016 | Team Helios | Design 1 |  |
| 2.0 | 04/08/2016 | Team Helios | Design 2 | Edit database schema, Add Poster |
|  |  |  |  |  |
|  |  |  |  |  |

TABLE OF CONTENTS

[1. High Level Design 3](#_Toc447919592)

[2. Low Level Design 4](#_Toc447919593)

[2.1 Timing Diagrams 4](#_Toc447919594)

[2.1.1 Client Settings Change 4](#_Toc447919595)

[2.1.2 Sensor Device Activation 5](#_Toc447919596)

[2.1.3 Current Monitor 6](#_Toc447919597)

[2.1.4 Control Device Activation 7](#_Toc447919598)

[2.2 Database schema 8](#_Toc447919599)

[2.3 UI Navigation tree 12](#_Toc447919600)

[2.4 Edit schedule communication diagram 13](#_Toc447919601)

[2.5 Server communications 14](#_Toc447919602)

[2.6 Usage graph request sequence diagram 16](#_Toc447919603)

[2.7 State Diagram of Client Devices 17](#_Toc447919604)

[3. Assumptions, Risks, & Benefits 18](#_Toc447919605)

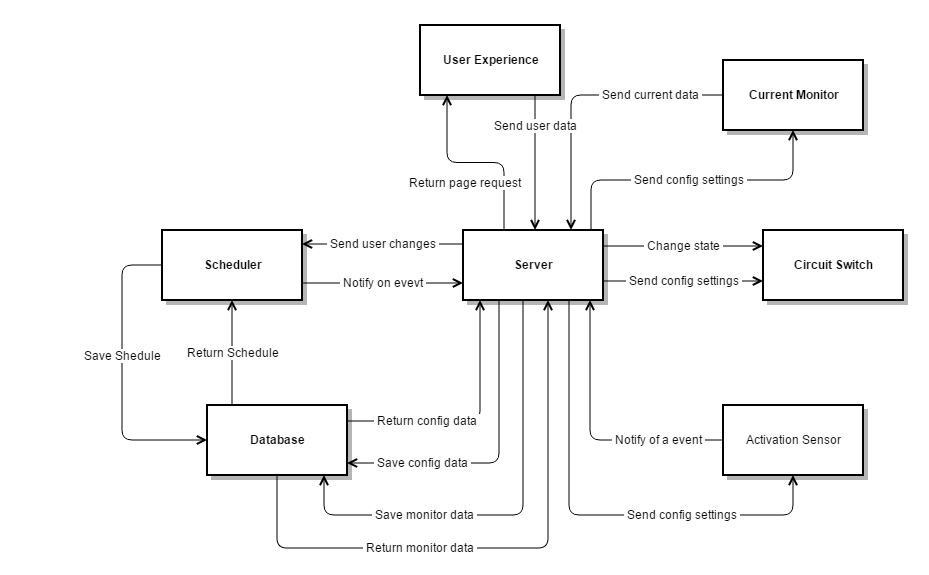
[3.1 ASSUMPTIONS 18](#_Toc447919606)

[3.2 RISKS 18](#_Toc447919607)

[3.3 BENEFITS 18](#_Toc447919608)

[4. Poster Draft 19](#_Toc447919609)

# 1. High Level Design

****

**Server:** The server is the central command and control of the system. The server will serve page requests and receive data from the user experience. The data received is processed or passed on to the proper controller. The server will log current monitor data and prepare the data for storage. The server will respond to notifications from the activation sensors and scheduler. The server will command the circuit switches to change states based on notifications.

**User Experience (GUI**): The user experience serves as the link between the user and the rest of the system. The user experience will display user requested information retrieved from the server. The user experience will also send user data, schedules and configuration, to the server for processing.

**Scheduler:** The scheduler monitors the current schedule for upcoming events and notifies the server on an event. The scheduler processes user changes passed by the server. The scheduler maintains the schedule stored on the database.

**Database:** Store and send data to the server and scheduler.

**Current Monitor:** Send average RMS wattage usage over a given time period to the server.

**Circuit Switch:** Change the state of the circuit based on commands received from the server.

**Activation Sensor:** Notify the server of sensor activation.

# 2. Low Level Design

## 2.1 Timing Diagrams

### 2.1.1 Client Settings Change



**User:** End user of Helios system. The user is expected to know how to change the available settings and the effects of those changes.

**Web Browser:** Any web browser capable of running JavaScript code and rendering HTML pages.

**Client Web Server:** A server application on the client device which can stream webpages to connected browsers. The web server will also handle stream data connections and requests from JavaScript in browsers.

**Client Filesystem:** Local filesystem of client device. The filesystem will hold any webpages and script code which the web server may need to host as well as storing any logging data the user wishes to maintain.

### 2.1.2 Sensor Device Activation



**User:** End user of Helios system.

**Sensor Device:** A Helios client device with a sensor unit attached. When the sensor is triggered by a user an acknowledgement is sent to the main Helios server along with any data generated by the sensor or being tracked by the sensor device.

**Control Device:** A Helios client device with a mains AC line voltage switch built in.

**Server:** The main Helios server. Filters and rebroadcasts messages to keep the various devices across the full system synchronized.

**Database Controller:** Logs and retrieves data for metrics and analysis.

### 2.1.3 Current Monitor



**Sensor Microcontroller:** Microcontroller which takes measurements from line current sensor and calculates RMS current from it.

**Line Current Sensor:** Measures AC current flowing through an attached sensor line,

**Wi-Fi Module:** Accessory device which will allow the main sensor microcontroller to communicate with the Helios server.

**Server:** The main Helios server. Filters and rebroadcasts messages to keep the various devices across the full system synchronized.

**Database Controller:** Logs and retrieves data for metrics and analysis.

### 2.1.4 Control Device Activation

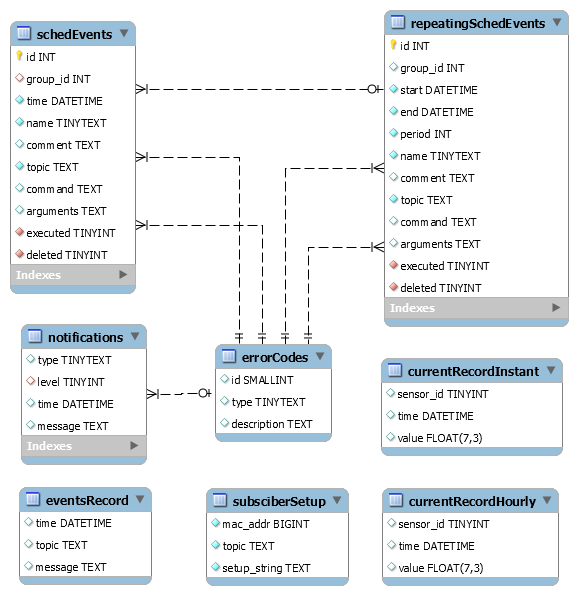


**Control Device:** A Helios client device with a control unit attached. When the device is signaled it will toggle power status of the attached control unit.

**Server:** The main Helios server. Filters and rebroadcasts messages to keep the various devices across the full system synchronized.

**Database Controller:** Logs and retrieves data for metrics and analysis.

## 2.2 Database schema



**schedEvents:** Stores single events as needed for execution as an MQTT command by the scheduler system.

* id - Auto-generated ID number, primary key
* group\_id - A group ID number to allow any modification or deletions to a single event to be propagated across any other events if this is a repeated event
* time – The time the event is to be triggered
* name – User assigned name
* comment – Any comment the user wants associated with this event
* topic – MQTT topic the event command is to be published on
* command – MQTT payload to be published when event is triggered
* arguments – Any arguments to be used by scheduler system
* executed – Detect whether or not the event has already been executed
* deleted – Flag for event deletion

**repeatingSchedEvents:** Stores events which need to be repeated on a period. These events will be consumed by the scheduler system and converted in a set of individual and uniquely timed events in the schedEvents table.

* id - Auto-generated ID number, primary key
* group\_id - A group ID number to allow any modification or deletions to a group of events to be propagated across any other events
* start – The beginning of the time range the events are to be scheduled
* end – The ending of the time range the events are to be scheduled
* period – The frequency in seconds the event is to be repeated
* name – User assigned name
* comment – Any comment the user wants associated with this set of events
* topic – MQTT topic the event command is to be published on
* command – MQTT payload to be published when event is triggered
* arguments – Any arguments to be used by scheduler system
* executed – Detect whether or not the scheduler system has created individual entries in the schedEvents table as described by this record
* deleted – Flag for event deletion

**errorCodes:** Stores any flags used by the system

**subscriberSetup:** Stores information to set up client subscriber devices so that system can persist in event of failure.

* mac\_addr – Mac addresses of client devices. Used to identify clients which have previously connected so any MQTT topics or local device settings can be resent to them. Absence of a record would thus indicate a new device in need of setup.
* topic – MQTT topic which client device should subscribe to
* setup\_string – String to send to clients to describe setup if necessary

**eventsRecord:** Stores a history of events sent by any systems which should be recorded. Primary use is for debugging.

* time – Time event was detected by recording system
* topic – MQTT topic event was detected on, if applicable
* message – message body of event, if applicable

**currentRecordInstant:** Stores a history of current sensor readings.

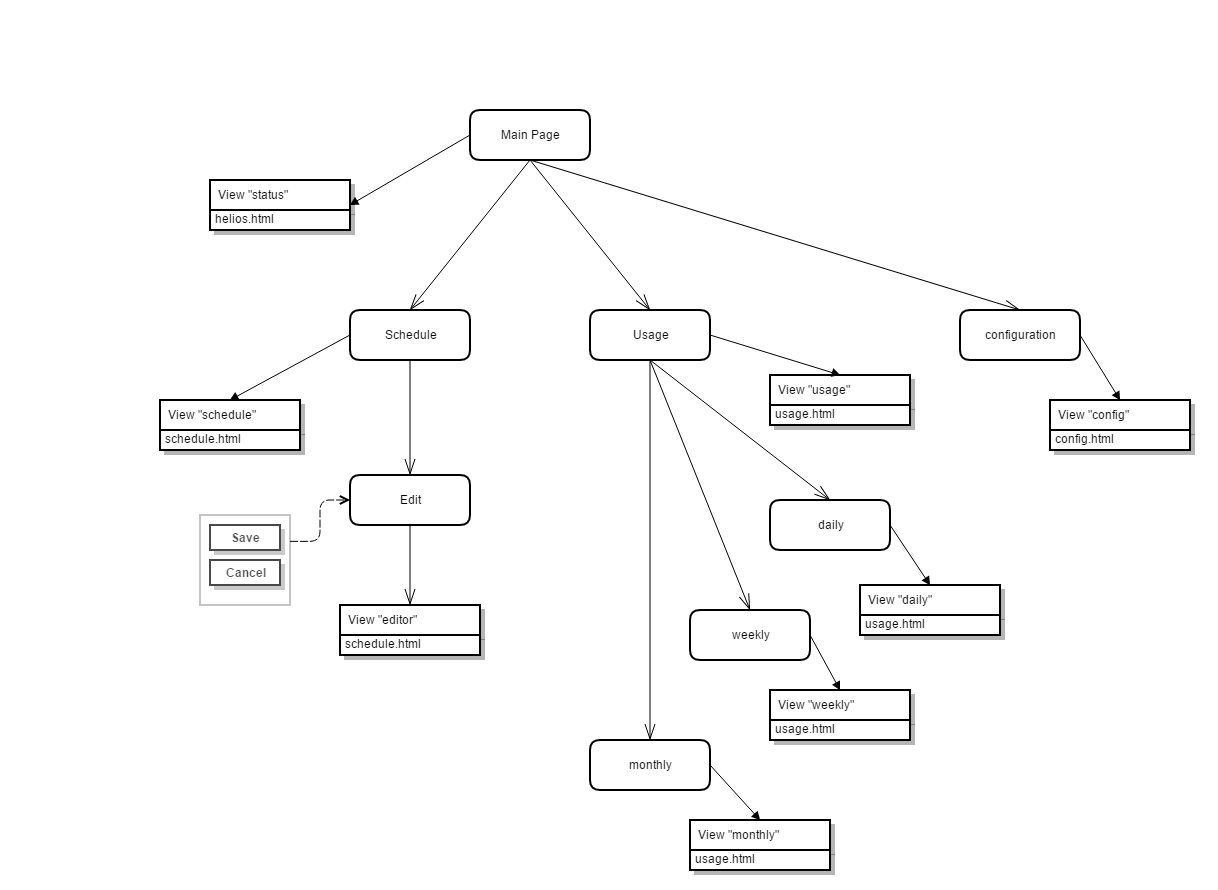
* sensor\_id – Unique ID of a CT current sensor feeding into the record system
* time – Time the reading was detected at
* value – The actual reading the CT sensor reported

**currentRecordHourly:** Stores an hourly average of current sensor readings to allow for easier calculation on the web browser portal.

**notifications:** Stores a record of notifications created by all parts of the Helios system which the user my wish or need to see.

* type – Describes the type of notification. e.g.: “error”, “warning”, or “message”
* level – Indicates the urgency level of the notification
* time – The time the notification was created
* message – The body of the message

## 2.3 UI Navigation tree



**Main page:** displays current circuit states and current energy usage. Three navigation buttons schedule, usage, and configuration redirect to the associated pages.

**Schedule:** displays the current schedule. One button, edit, allows interaction with the schedule table.

**Edit:** table values change on click. Two buttons save and cancel, save sends the server the edited table to save and cancel returns the user to the schedule view.

**Usage:** display 6-month usage graph. Three buttons daily, weekly, and monthly.

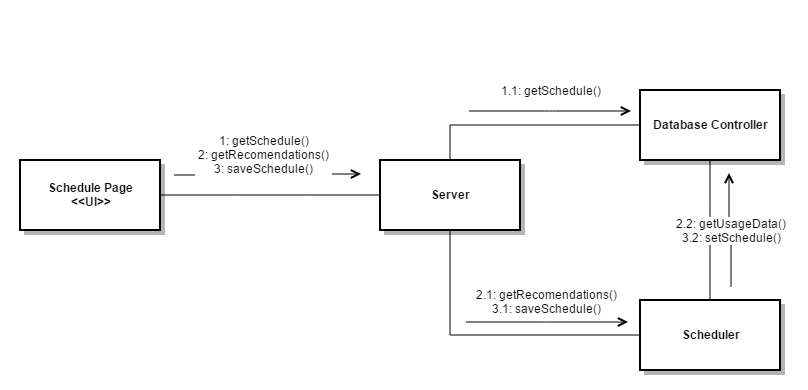
**Daily:** display 1-day usage graph. Three buttons 6 months, weekly, and monthly.

**Weekly:** display 1-week usage graph. Three buttons 6 months, daily, and monthly.

**Monthly:** display 1-month usage graph. Three buttons 6 months, daily, and weekly.

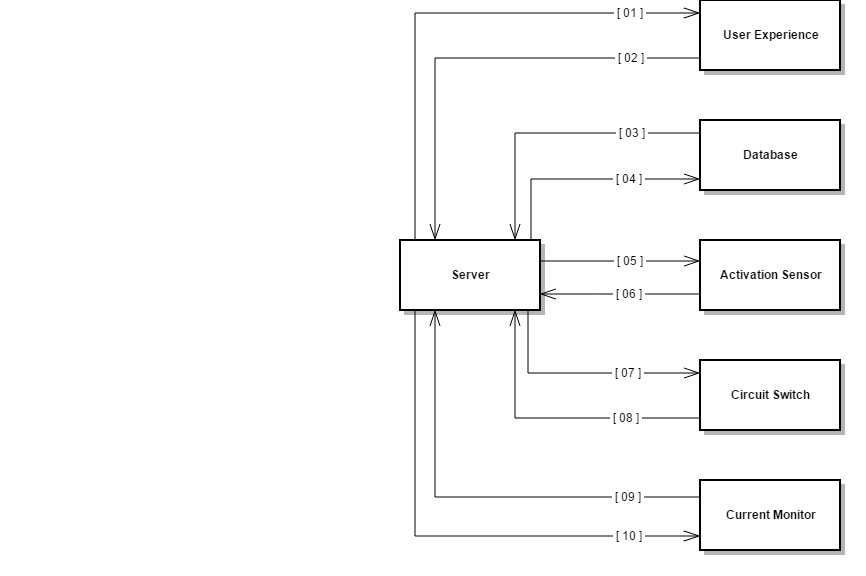
**Configuration:** display current configuration setting for the system.

## 2.4 Edit schedule communication diagram



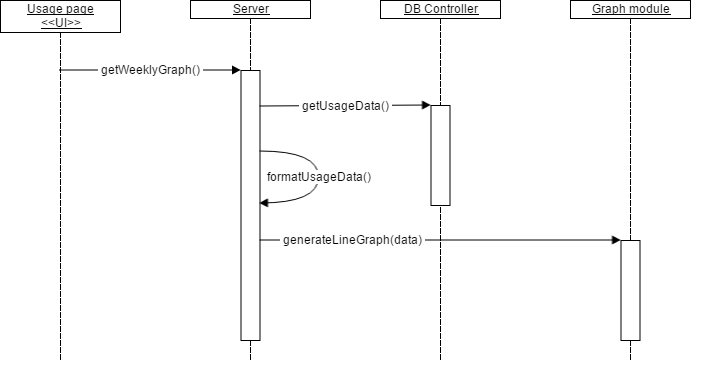
1. Page asks the server for the current schedule
   1. Server makes a request to the database controler for the current schedule. Returns data
2. Page asks server for the recommendation set
   1. Server asks the scheduler for the recommendation set
   2. Scheduler requests usage data from database controler. Generats recommendations and return data
3. Page sends schedule data to be saved to the server
   1. Server passes schedule data to the scheduler
   2. Schedular formats schedule data and request the data base updat the new schedule

## 2.5 Server communications



1. **Communication from server to user experience**
   1. Update current energy usage value on a 5 second interval
   2. Generate and send usage graphs requested by user experience
   3. Retrieve and send schedule table to user experience
   4. Send configuration settings to the user experience
2. **Communication from user experience to server**
   1. Get request for current usage data
   2. Get request for usage graph
   3. Get request for current schedule
   4. Send server edited schedule for storage
   5. Get request for current configuration data
3. **Communication from database to server**
   1. Return query results
4. **Communication from server to data base**
   1. Query table
   2. Send data to be saved to table
5. **Communication from server to activation switch**
   1. Send configuration data
6. **Communication from activation switch to server**
   1. Notify a activation of the sensor
   2. Request configuration data
7. **Communication from server to circuit switch**
   1. Send state change command
   2. Send configuration data
8. **Communication from circuit switch to server**
   1. Notify change of circuit state
   2. Request configuration data
9. **Communication from current monitor to server**
   1. Send current energy usage on a 5 second interval
   2. Request configuration data
10. **Communication from server to current monitor**
    1. Send configuration data

## 2.6 Usage graph request sequence diagram



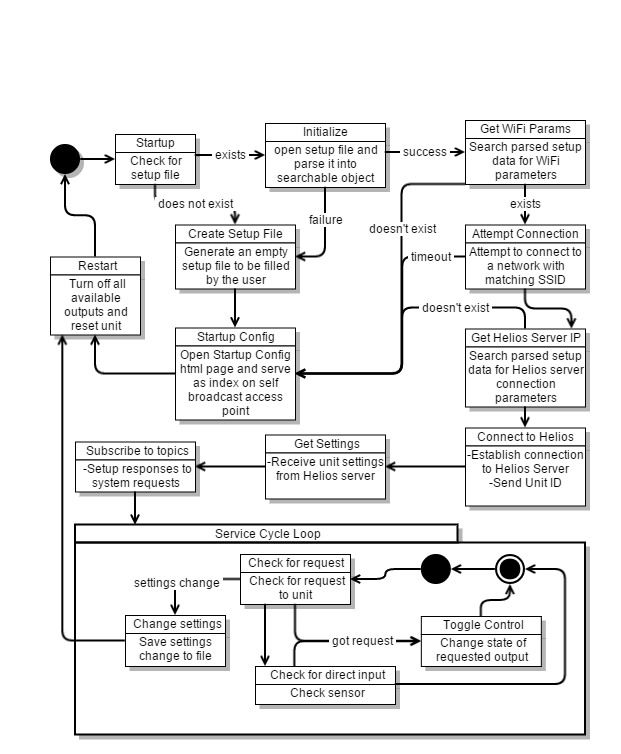
**getWeeklyGraph():** Request the server for a PNG file of a line graph depicting the energy use over the previous week.

**getUsageData():** Queries the database for the energy usage data.

**formatUsageData():** Formats the queried usage data into a form usable by the graphing module.

**generateLineGraph(data):** Generates a line graph from the formatted data.

## 2.7 State Diagram of Client Devices



# 3. Assumptions, Risks, & Benefits

## 3.1 ASSUMPTIONS

The following is a list of assumptions:

* User has a wireless network
* Network can transmit Helios messages without saturating bandwidth
* Ignore compliance issues
* 120 V AC power
* User has privileges to set private IP addresses
* Simulate 6 months of usage data

## 3.2 RISKS

The following is a list of risks:

* Schedule very aggressive
* Team lacks access to advanced sensor devices
* Team lacks experience working with AC mains lines
* Use of wireless chips for communication

## 3.3 BENEFITS

The following is a list of benefits:

* User can track electrical usage over time. Allowing the user to make informed decisions on current electrical usage
* Schedule allows the user to completely close a circuit during set times of day. Allowing the user to conserve energy when the room is not in use
* Schedule suggests times to close a circuit. Providing the user with guidance on how to conserve energy
* The system will close circuits automatically based on a set of rules and sensor information. This automation will reduce overall energy use.

# 4. Poster Draft

